

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	571	386/55	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L2	0	L1 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L3	1802	345/629	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L4	1971	345/619	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L5	3421	L4 or L3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L6	2	345/619-629	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L7	5	L5 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48

EAST Search History

L8	357	(E-card\$1 (E near2 card\$1)) & (((add\$3 annotat\$3) near3 link\$1 URL))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L9	8	L8 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L10	2	("5337362").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L11	131050	345/619, "629"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L12	2	("66208436" "5880740").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L13	479	715/512	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L14	197	L13 & (((add\$3 annotat\$3) near2 link\$1 URL))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48

EAST Search History

L15	1	L14 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L16	32	("5337362" "5444779" "5495581" "5640193" "5804803" "5838313" "5838458" "5880740" "5893126" "5905248" "6076734" "6208436" "6256638" "6448979" "6572661" "6674923").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L17	33	L15 or L16	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L18	5	L17 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L19	511	L13 or L16	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L20	2	("6448979").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L21	0	345/619,629	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48

EAST Search History

L22	100	715/741	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L23	1	L22 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L24	0	L13 & L16	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L25	0	L1 & (E-card\$1 (E near2 card\$1)) & (((add\$3 annotat\$3) near3 link\$1 URL))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L26	2	("6208436").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L27	4	L16 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L28	0	L14 & (multimedia same link\$3 same ((paper physical printted) adj (document\$1 media medium object\$1)))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48

EAST Search History

L29	198	L13 & (((add\$3 annotat\$3) near3 link\$1 URL))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L30	4739	358/474	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L31	0	L30 & (multimedia same link\$3 same (paper physical printted))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 09:48
L32	2	("5337362").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 10:05
L33	11	("5692073" "6081261" "6326946" "6330976" "6356923").PN.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/19 10:05



[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide

the origami project +physical document -link +mark up -docu

SEARCH

THE ACM DIGITAL LIBRARY

[Advanced Search](#)

[? Search
Tips](#)

Enter words, phrases or names below. Surround phrases or full names with double quotation marks.

Search within Results: 17 found

the origami project +physical
document -link +mark up -document
+annotation multimedia +web server
+pointer



[Clear result set](#)

SEARCH

Desired Results:

must have all of the words or phrases

must have any of the words or phrases

must have none of the words or phrases

Name or Affiliation:

Authored by: all any none

Edited by: all any none

Reviewed by: all any none

SEARCH

Only search in:*

Title Abstract Review All Information

*Searches will be performed on all available information, including full text where available, unless specified above.

ISBN / ISSN: Exact Expand

DOI: Exact Expand

SEARCH

Published:

By: all any none

In: all any none

Since:

Month Year

Before:

Month Year

As: Any type of publication

Conference Proceeding:

Sponsored By:

Conference Location:

Conference Year:

yyyy

SEARCH**Classification: (CCS)** Primary OnlyClassified as: all any noneSubject Descriptor: all any noneKeyword Assigned: all any none**Results must have accessible:** Full Text Abstract Review**SEARCH**

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

 **PORTAL**
USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: The ACM Digital Library The Guide
 the origami project +physical document -link +mark up -docu

THE ACM DIGITAL LIBRARY[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

[the origami project](#) [physical document](#) [link](#) [mark up](#) [document](#) [annotation](#) [multimedia](#) [web server](#) [pointer](#)Sort results by [relevance](#) [Save results to a Binder](#)[Try an Advanced Search](#)Display results [expanded form](#) [Search Tips](#)[Try this search in The ACM Guide](#) [Open results in a new window](#)

Results 1 - 17 of 17

Relevance scale

- 1 [Visual information and collaboration: Improving interpretation of remote gestures with telepointer traces](#)

 Carl Gutwin, Reagan PennerNovember 2002 **Proceedings of the 2002 ACM conference on Computer supported cooperative work CSCW '02****Publisher:** ACM PressFull text available:  [pdf\(355.51 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Gestural communication is an important part of shared work, both in face-to-face settings and distributed environments. However, gestures in groupware are often difficult to see and interpret because of disruptions to their motion caused by network jitter. One way to improve the visibility of remote gestures is by using traces-visualizations of the last few moments' of a remote pointer motion. We carried out an experiment to test the effectiveness of traces in helping people interpret gestures ...

Keywords: consequential communication, gesture, groupware usability, jitter, network delay, telepointer traces

- 2 [Virtual and augmented reality: FingARtips: gesture based direct manipulation in Augmented Reality](#)

 Volkert Buchmann, Stephen Violich, Mark Billinghurst, Andy CockburnJune 2004 **Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia GRAPHITE '04****Publisher:** ACM PressFull text available:  [pdf\(590.58 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a technique for natural, fingertip-based interaction with virtual objects in Augmented Reality (AR) environments. We use image processing software and finger- and hand-based fiducial markers to track gestures from the user, stencil buffering to enable the user to see their fingers at all times, and fingertip-based haptic feedback devices to enable the user to feel virtual objects. Unlike previous AR interfaces, this approach allows users to interact with virtual content using ...

Keywords: Augmented Reality, gesture interaction, occlusion

3 Flow-insensitive type qualifiers

 Jeffrey S. Foster, Robert Johnson, John Kodumal, Alex Aiken
November 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volu
28 Issue 6

Publisher: ACM Press

Full text available:  pdf(910.93 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe flow-insensitive type qualifiers, a lightweight, practical mechanism for specifying and checking properties not captured by traditional type systems. We present a framework for adding new, user-specified type qualifiers to programming languages with static type systems, such as C and Java. In our system, programmers add a few type qualifier annotations to their program, and automatic type qualifier inference determines the remaining qualifiers and checks the annotations for consistency ...

Keywords: Type qualifiers, const, constraints, security, static analysis, taint, types

4 Eraser: a dynamic data race detector for multithreaded programs

 Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro, Thomas Anderson
November 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 4

Publisher: ACM Press

Full text available:  pdf(136.04 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Multithreaded programming is difficult and error prone. It is easy to make a mistake in synchronization that produces a data race, yet it can be extremely hard to locate this mistake during debugging. This article describes a new tool, called Eraser, for dynamically detecting data races in lock-based multithreaded programs. Eraser uses binary rewriting techniques to monitor every shared-memory reference and verify that consistent locking behavior is observed. We present several case studies ...

Keywords: binary code modification, multithreaded programming, race detection

5 An analysis of a resource efficient checkpoint architecture

 Haitham Akkary, Ravi Rajwar, Srikanth T. Srinivasan
December 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Is
4

Publisher: ACM Press

Full text available:  pdf(757.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Large instruction window processors achieve high performance by exposing large amounts of instruction level parallelism. However, accessing large hardware structures typically required to buffer and process such instruction window sizes significantly degrade the cycle time. This paper proposes a novel checkpoint processing and recovery (CPR) microarchitecture, and shows how to implement a large instruction window processor without requiring large structures thus permitting a high clock frequency ...

Keywords: Computer architecture, checkpoint architecture, high-performance computing, scalable architecture

6 Incremental and demand-driven points-to analysis using logic programming

 Diptikalyan Saha, C. R. Ramakrishnan
July 2005 **Proceedings of the 7th ACM SIGPLAN international conference on Principles and practice of declarative programming PPDP '05**
Publisher: ACM Press

Full text available: [pdf\(225.95 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Several program analysis problems can be cast elegantly as a logic program. In this paper we show how recently-developed techniques for incremental evaluation of logic programs can be refined and used for deriving practical implementations of incremental program analyzers.

Incremental program analyzers compute the changes to the analysis information due to small changes in the input program rather than re-analyzing the program. Demand-driven analyzers compute only the information requested by th ...

Keywords: demand-drive analysis, incremental analysis, logic programming, pointer analysis

7 An evaluation of staged run-time optimizations in DyC

Brian Grant, Matthai Philipose, Markus Mock, Craig Chambers, Susan J. Eggers

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation PLDI '99**, Volume 34 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.54 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Previous selective dynamic compilation systems have demonstrated that dynamic compilation can achieve performance improvements at low cost on small kernels, but they have had difficulty scaling to larger programs. To overcome this limitation, we developed DyC, a selective dynamic compilation system that includes more sophisticated and flexible analyses and transformations. DyC is able to achieve good performance improvements on programs that are much larger and more complex than the kernels. We ...

8 Session 2: An empirical study of the robustness of MacOS applications using random testing

Barton P. Miller, Gregory Cooksey, Fredrick Moore

July 2006 **Proceedings of the 1st international workshop on Random testing RT '06**

Publisher: ACM Press

Full text available: [pdf\(165.21 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We report on the fourth in a series of studies on the reliability of application programs in the face of random input. Over the previous 15 years, we have studied the reliability of UNIX command and X-Window based (GUI) applications and Windows applications. In this study, we apply our fault testing techniques to applications running on the Mac OS X operating system. We continue to use a simple, or even simplistic technique: unstructured black-box random testing, considering a failure to be ...

Keywords: fuzz, random testing

9 Posters--visualization: v4v: a View for the Viewer

Mira Dontcheva, Steven M. Drucker, Michael F. Cohen

November 2005 **Proceedings of the 2005 conference on Designing for User eXperience DUX '05**

Publisher: AIGA: American Institute of Graphic Arts

Full text available: [pdf\(431.73 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

We present a **View for the Viewer** (v4v), a slide viewer that focuses on the needs of the viewer of a presentation instead of the presenter. Our design centers on representing the deck of slides as a stack embedded in a 3-D world. With only single button clicks, the viewer can quickly and easily navigate the deck of slides. We provide four types of annotation techniques and have designed a synchronization mechanism that makes it easy for the viewer to move in and out of sync with the presentation ...

Keywords: note taking, presentations, slide representation, user interface design, visualization

10 Articles: A context-aware methodology for very small data base design

 C. Bolchini, F. A. Schreiber, L. Tanca
March 2004 **ACM SIGMOD Record**, Volume 33 Issue 1

Publisher: ACM Press

Full text available:  pdf(381.70 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The design of a Data Base to be resident on portable devices and embedded processors for professional systems requires considering both the device memory peculiarities and the mobility aspects, which are an essential feature of the embedded applications. Moreover, these devices are often part of a larger Information System, comprising fixed and mobile resources. We propose a complete methodology for designing Very Small Data Bases, from the identification of the device resident portions down to ...

11 Mix ten years later

 Neil D. Jones
June 1995 **Proceedings of the 1995 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation PEPM '95**

Publisher: ACM Press

Full text available:  pdf(1.76 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 Patterns: Micro pattern evolution

 Sunghun Kim, Kai Pan, E. James Whitehead
May 2006 **Proceedings of the 2006 international workshop on Mining software repository MSR '06**

Publisher: ACM Press

Full text available:  pdf(654.17 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

When analyzing the evolution history of a software project, we wish to develop results that generalize across projects. One approach is to analyze design patterns, permitting characteristic of the evolution to be associated with patterns, instead of source code. Traditional design patterns are generally not amenable to reliable automatic extraction from source code, yet automation is crucial for scalable evolution analysis. Instead, we analyze "micro pattern" evolution; patterns whose abstract ...

13 Animation: SnakeToonz: a semi-automatic approach to creating cel animation from video

 Aseem Agarwala
June 2002 **Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering NPAR '02**

Publisher: ACM Press

Full text available:  pdf(639.81 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

SnakeToonz is an interactive system that allows children and others untrained in cel animation to create two-dimensional cartoons from video streams and images. The ability to create cartoons has traditionally been limited to professional animation houses and trained artists. SnakeToonz aims to give anyone with a video camera and a computer the ability to create compelling cel animation. This is done by combining constraints of the cartooning medium with simple user input and analysis of that in ...

14 A model of keyboard configuration requirements

 Shari Trewin, Helen Pain
January 1998 **Proceedings of the third international ACM conference on Assistive technologies Assets '98**

Publisher: ACM Press

Full text available: [txt\(44.76 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: Bounce Keys, Repeat Keys, Sticky Keys, empirical studies, keyboard configuration, keyboards, motor disabilities, user modelling

15 Query processing for relational data: Supporting ad-hoc ranking aggregates

 Chengkai Li, Kevin Chen-Chuan Chang, Ihab F. Ilyas

June 2006 **Proceedings of the 2006 ACM SIGMOD international conference on Management of data SIGMOD '06**

Publisher: ACM Press

Full text available: [pdf\(344.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents a principled framework for efficient processing of ad-hoc *top-k* (ranking) aggregate queries, which provide the *k* groups with the highest aggregates as results. Essential support of such queries is lacking in current systems, which process the queries in a naïve materialize-group-sort scheme that can be prohibitively inefficient. Our framework is based on three fundamental principles. The Upper-Bound Principle dictates the requirements of early pruning, and ...

Keywords: OLAP, aggregate query, decision support, ranking, top-k query processing

16 Static single assignment form for machine code

 Allen Leung, Lal George

May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation PLDI '99**, Volume 34 Issue 5

Publisher: ACM Press

Full text available: [pdf\(1.31 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Static Single Assignment (SSA) is an effective intermediate representation in optimizing compilers. However, traditional SSA form and optimizations are not applicable to programs represented as native machine instructions because the use of dedicated registers imposed by calling conventions, the runtime system, and target architecture must be made explicit. We present a simple scheme for converting between programs in machine code and in SSA, such that references to dedicated physical registers ...

17 Array morphology

 Robert Bernecky

September 1993 **ACM SIGART APL Quote Quad , Proceedings of the international conference on APL APL '93**, Volume 24 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.02 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Array morphology is the study of the form, structure, and evolution of arrays. An *array annotator* for a program written in an applicative array language is an abstract syntax tree for the program amended with information about the arrays created by that program. Array notations are useful for the production of efficient compiled code for applicative array programs. Array morphology is shown to be an effective compiler writer's tool. Examples of an array annotator in action are pre-

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)